

Lab 01 Report - Introduction to Open Source Software

Personal Information

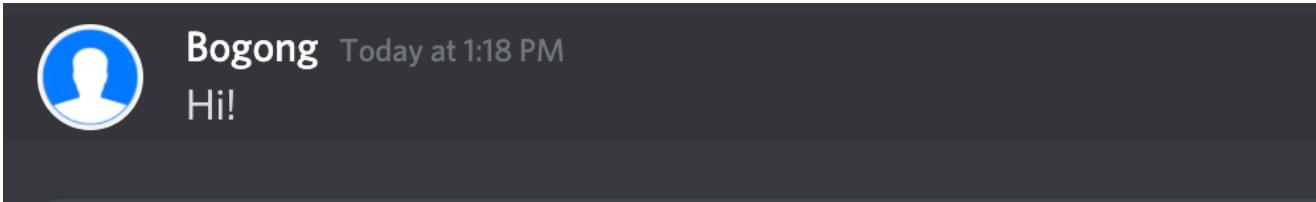
Bogong Yang



CS senior undergrad

Discord handle: deadbeef#6255, appears as Bogong in the OSS channel

Screenshot of discord:

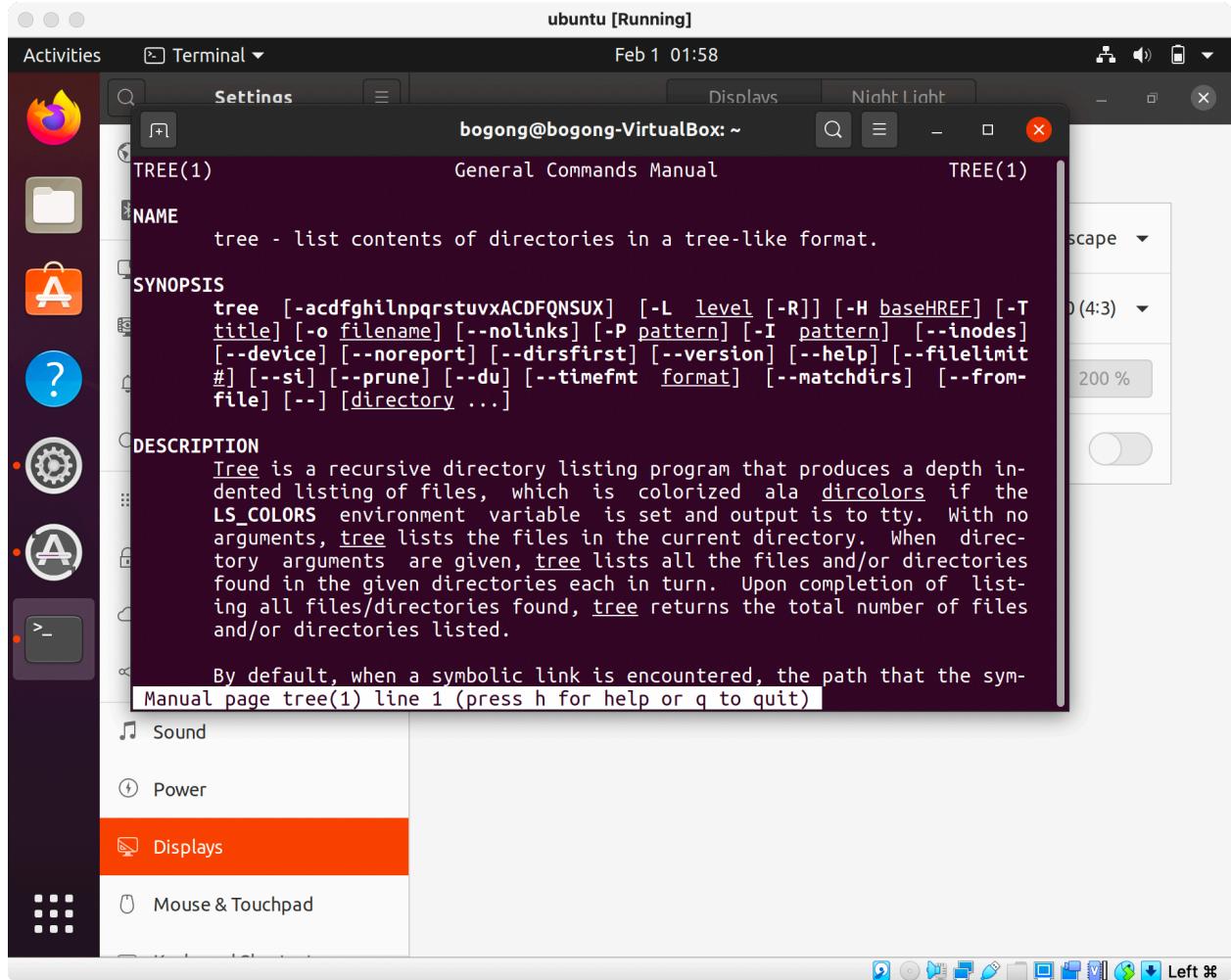


Reading Assignments

- More suggestions for smart questions:
 1. Provide a simple, short code snippet that replicates the problem instead of posting hundreds of lines of code
 2. When asking a question on forums such as StackOverflow, if someone thinks that the question is a duplicate, be sure to explain how the question differs from the previous ones precisely
- Chapter 3 of Free Culture
 - I do not know much about American copyright laws therefore I cannot judge who was right on the case, but I do think potential legal problems restricts creativity in some ways. I imagine if the search engine were to be open-source and properly licensed, the student might hold against the lawsuits. However, as the passage suggests, even if the student won, the cost would still be too high for the student to pay. In this case, law was used as a tool by RIAA for making profit, as well as a restriction posed on the student researcher. I feel like unless there is some modifications in the "procedural justice" law system, such threats will still be lingering around. On the other hand, if the music were to be "open-source" in some way, RIAA might not even existed from the beginning, but the commercial nature of the market could not be simply ignored.

Linux

- Screenshot of `man tree`



Regex

Practice Problems:

1. Q1:

Exercise 1: Matching Numbers

Task Text

Match 3.14529



Match -255.34



Match 128



Match 1.9e10



Match 123,340.00



Skip 720p

`^-*\d+.*e*\d+$`

Continue >

2. Q2:

Exercise 2: Matching Phone Numbers

Task	Text	Capture Groups
Capture	415-555-1234	415
Capture	650-555-2345	650
Capture	(416)555-3456	416
Capture	202 555 4567	202
Capture	4035555678	403
Capture	1 416 555 9292	416

(\d\d\d)

Continue >

3. Q3:

Exercise 3: Matching Emails

Task	Text	Capture Groups
Capture	tom@hogwarts.com	tom
Capture	tom.riddle@hogwarts.com	tom.riddle
Capture	tom.riddle+regexone@hogwarts.com	tom.riddle
Capture	tom@hogwarts.eu.com	tom
Capture	potter@hogwarts.com	potter
Capture	harry@hogwarts.com	harry
Capture	hermione+regexone@hogwarts.com	hermione

((\w)*\.\?(\w)*)

Continue >

4. Q4:

Exercise 4: Capturing HTML Tags

Task	Text	Capture Groups
Capture	<a>This is a link	a
Capture	Link	a
Capture	<div class='test_style'>Test</div>	div
Capture	<div>Hello world</div>	div

<(\w+)|

Continue >

5. Q5:

Exercise 5: Capturing Filename Data

Task	Text	Capture Groups
Skip	.bash_profile	
Skip	workspace.doc	
Capture	img0912.jpg	img0912 jpg
Capture	updated_img0912.png	updated_img0912 png
Skip	documentation.html	
Capture	favicon.gif	favicon gif
Skip	img0912.jpg.tmp	
Skip	access.lock	
<code>(.+)\.(jpg png gif)\$</code>		Continue >

6. Q6:

Exercise 6: Matching Lines

Task	Text	Capture Groups
Capture	The quick brown fox...	The quick brown fox...
Capture	jumps over the lazy dog.	jumps over the lazy dog.
<code>((\w+\s?\.*)+)</code>		Continue >

7. Q7:

Exercise 7: Extracting Data From Log Entries

Task	Text	Capture Groups
Skip	W/dalvikvm(1553): threadid=1: uncaught exception	
Skip	E/(1553): FATAL EXCEPTION: main	
Skip	E/(1553): java.lang.StringIndexOutOfBoundsException	
Capture	E/(1553): at widget.List.makeView(ListView.java:1727)	makeView ListView.java 1727
Capture	E/(1553): at widget.List.fillDown(ListView.java:652)	fillDown ListView.java 652
Capture	E/(1553): at widget.List.fillFrom(ListView.java:709)	fillFrom ListView.java 709
<code>\.(\w+)\((\w+\.\w+):(\d+)\)\$</code>		Continue >

Crosswords:

1. Q1:

[^SPEAK]+	EP IP EF
HE LL O+	H E
[PLEASE]+	L P

2. Q2:

(AB OE SK)	(A B C) \ 1
. *M?O.*	B O
(AN FE BE)	B E

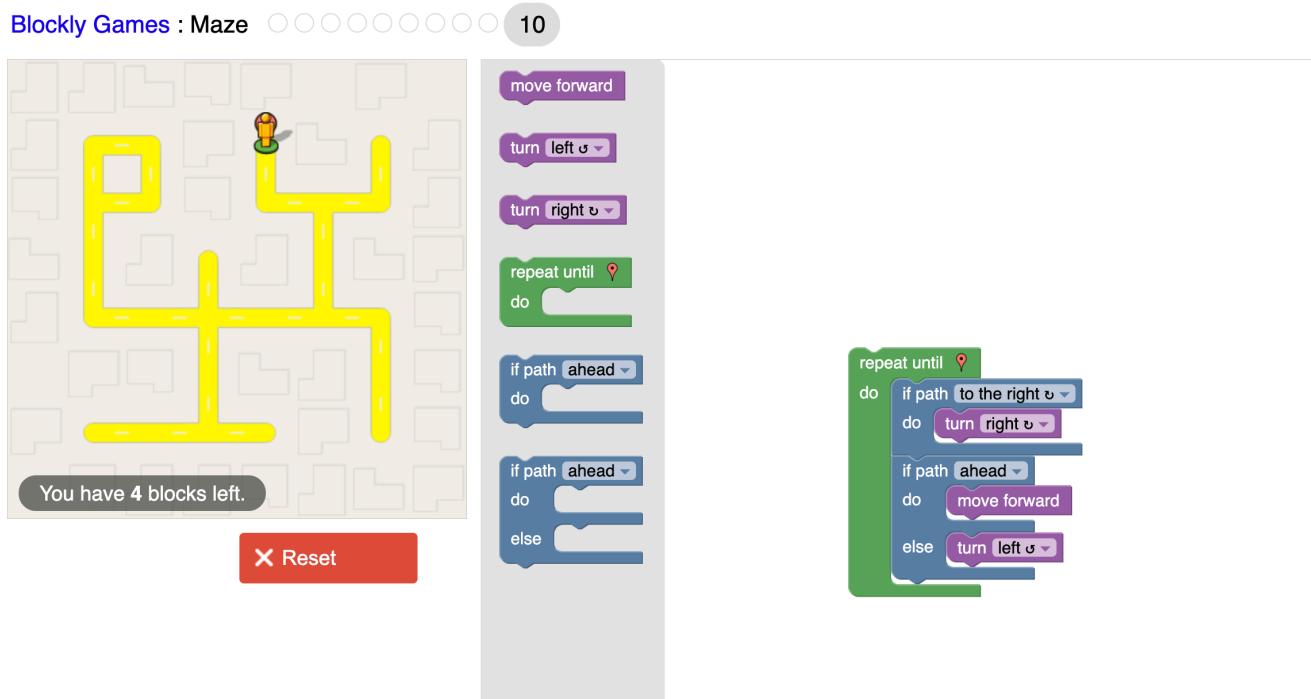
3. Q3:

$(AB 0 OR)^+$	
$[COBRA]^+$	
$(.)^+\backslash 1$	O O
$[^ABRC]^+$	O O

4. Q4:

	.	?	.	+
	+			+
$[*]^+$	*		*	
/	/		/	

Blockly



Reflection:

I am thinking of one of my own projects, and it was meant to be an online platform where faculties can post URP positions and students can apply to them on the platform. The problem at hand is that professors and faculties are always busy, and the emails sent by students may simply sink into the tons of emails in the inbox and remain unread. Also, from my most recent experiences, I am considering to build a platform such that students could request professors for writing recommendation letters (with notification features).